EECS442 WN24 Practice Midterm

Question 1.

Given a 5x5 image and a 3x3 filter, perform the convolution operation to detect vertical edges within the image. The image and filter are provided below:

Image:

10	10	10	10	10
10	20	20	20	10
10	20	30	20	10
10	20	20	20	10
10	10	10	10	10

Filter:

1	0	-1
2	0	-2
1	0	-1

-1 0 1
-2 0 2
-1 0 1

a. (9 points) Apply the filter to the image using convolution. Assume zero-padding is not used, so the output will be a 3x3 image. Provide the result.

-10 + 10 = 0
-20 + 40 = 20.
-10 + 30 = 20.

40.

40 0 -40
-60
60 0
40 0 -40.

b. (3 points) Explain briefly how the values in the resulting image relate to vertical edge detection. Why do positive or negative values appear, and what do they indicate about the presence of edges?

vertical edge.

possitive: the left side of the edge.
negative the. right side of the edge.
the density of the edge is higher than the background

Question 2.

Let's consider a simple convolutional neural network, which takes a **1x64x64 image** as input. We use this network to perform a **binary** image classification task. Part of the model architecture is given below. Conv represents the convolutional layer, its shape is in the form of (number of filters, height, width). Shape of the MaxPooling layer is in the form of (height, width).

The padding size of the convolution layer is 1x1. Step sizes for convolution layers are all 1x1. No padding is applied to Pooling layers. Step sizes for pooling layers are the same as the pooling kernel.

128x64x64 3x3x1x128

Layer Name	Layer Shape	Output Shape	#Parameters	
Conv-1	(128, 3, 3)	Shape_1	Count_1	
ReLU-1	-	128x64x64 Shape_2	Count_2	0
MaxPooling-1	(2, 2)	128x32x32 Shape_3	Count_3	0
Global Average Pooling	-	128x1x1 Shape_4	Count_4	0
Fully-Connected-1	Shape_5	~~2x1x1~~ Shape_6 1	Count_5 128	~~128x2~~ 128

a) (5 points) Decide the output shape of each layer and complete all the Shape_X entries in the table.

(Hint: Global Average Pooling averages the pixels spatially and reduces the spatial size of any image to be 1x1.)

b) (5 points) Decide the number of parameters of each layer and complete all the Count_X entries in the table. If the layer contains no parameter, just write 0. **Ignore the bias term.**

Question 3.

Recall the second-moment matrix (M as shown below) which determines the corner-ness of a pixel using the gradients I_x and I_y of the image $I(x, y)$ where $I_x = \frac{\partial I(x,y)}{\partial x}$ and $I_y = \frac{\partial I(x,y)}{\partial y}$.

$$\boldsymbol{M} = \begin{bmatrix} \sum_{x,y \in W} I_x^2 & \sum_{x,y \in W} I_x I_y \\ \sum_{x,y \in W} I_x I_y & \sum_{x,y \in W} I_y^2 \end{bmatrix}$$

a) (5 points) Explain briefly how the eigen-values (say λ_1 and λ_2) of M determines the presence of a flat region, an edge and a corner in a window of an image.

flat : λ_1 : small λ_2 : small

edge : λ_1 small λ_2 large or λ_1 large λ_2 small

corner: λ_1 large λ_2 large.

b) (2 points) Explain how the eigen-values change when the brightness of the image is increased by a constant value 'b' as $I_{new} = I_{old} + b$ (Assume that the maximum pixel intensity of the new image I_{new} does not exceed the allowed range of intensities.)

since $I_x = \frac{\partial I(x,y)}{\partial x}$ the constant b will not influence the derivative. so the eigen-values will not change.

c) (3 points) In practice, the second-moment matrix is scaled using a Gaussian scheme as shown below. Give an intuitive reason for performing this weighing.

$$\boldsymbol{M} = \begin{bmatrix} \sum_{x,y \in W} w(x,y) I_x^2 & \sum_{x,y \in W} w(x,y) I_x I_y \\ \sum_{x,y \in W} w(x,y) I_x I_y & \sum_{x,y \in W} w(x,y) I_y^2 \end{bmatrix}$$

since we want to add weight for the pixel with different distance to the center point, the far you away from the center pixel. the less influence you will have on it.

Question 4.

The first image is the original, and image_1 and image_2 are transformed versions of the first.

a) (5 points) Identify whether a **rigid or affine** transformation was used to generate Image_1 from the Original Image. Identify the properties that are preserved and those that are not preserved in this transformation, specifically in terms of lines and parallelism. Additionally, state the degrees of freedom involved in this type of transformation.

affine.

since · lines are still lines.
· parallel lines are still lines.
· But origin· is not origion.

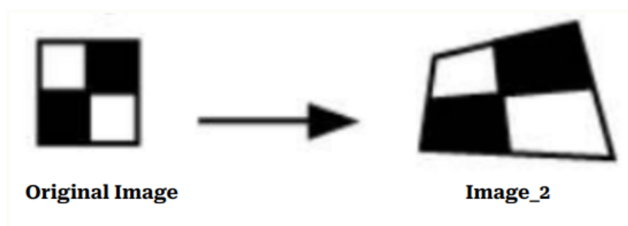
degree of freedom.

$$\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix}$$

if ~~we have~~ |H| = 1

then if ~~x~~ · 6

degree of freedom.



b) (5 points) Identify whether an **affine or perspective** transformation was used to generate Image_2 from the Original Image. Identify the properties that are preserved and those that are not preserved in this transformation, specifically in terms of lines and parallelism. Additionally, state the degrees of freedom involved in this type of transformation.

perspective. ,

since. · lines are still lines.
· parallel lines are not parallel lines.
· original may not be original.

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

let i=1 or |H| =1 ✓

we have 8 degree of freedom.

Question 5

a. (2 points) Write out 2 advantages of using batch normalization in neural networks.

①: avoide the backword proporgation result to be too big or too small.

②: can add more layer to design more complex nn architecture.

enable faster learning rate. → leads to faster model convergence.

b. (3 points) Write out 2 ways to prevent overfitting in the model and 1 way to prevent underfitting.

prevent overfitting:

①. add regularization term.

②. increase the training data set size.

reduce model complexity. Early stop.

prevent underfitting.

①. decrease the hyperparameter before the regularization term.

②. increase the learning rate & increase training data set.

using more complex model.

increase Epoch times.

c. (5 points) Given the set of points [(1, 3), (3, 5.5), (3, 7), (4, 9.75), (5, 11)], perform the RANSAC algorithm to fit a linear model. Use the following simplifications.

i. Use (1,3) and (5,11) as your "random" selection.
ii. Use a distance threshold of 1 for counting inliers.
iii. Calculate the distance between the point and the line as distance along the y-axis (difference in y-coordinates). (Instead of the shortest distance between the line and the point)

Write out the equation of the model, the distance of the points to the line, and the final sets of inliers and outliers.

Line $y = ax + b$. $y = 2x + 1$.

$3 = a + b$.

$11 = 5a + b$.

$4a = 8$

$a = 2 \quad b = 1$

distance from point to line. (y-axis).

(1, 3).

$y = 2 + 1 = 3$ distance = 0. inlier

(3, 5.5)

$y = 3 \times 2 + 1 = 7$ $7 - 5.5 = 1.5$. outlier.

(3, 7)

$y = 2 \times 3 + 1 = 7$ $7 - 7 = 0$. inlier.

(4, 9.75).

$y = 2 \times 4 + 1 = 9$ $9.75 - 9 = 0.75$ inlier.

(5, 11)

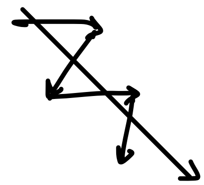
$y = 2 \times 5 + 1 = 11$ $11 - 11 = 0$ inlier

Question 6

(4 points) Explain what it means to use momentum for gradient descent. What is the benefit of using gradient descent with momentum?

momentum is a faster way to find the local / global minimum.

it will average the gradient



Benifit: faster.

more accurate.

less iteration time.

take a step towards the averaged gradient direction as opposed to the current one.

avoid noise. allows progressing to a good direction consistently ✓.

Question 7

a. (3 points) Explain the concept of regularization in machine learning or linear regression and explain when we need such regularization.

when we use the training dataset to train the model. sometime the model is too fit to the training data, the variety of the model is too high. and it will also have a bad performance when we do the test. So we need the regularization term to decrease the model complexity.

When our training error is really low. but the testing error is big.

b. (3 points) An example of regularization is adding a loss term on the model weights. Explain what effects we expect from the weight regularization and why is it a good idea for machine learning.

we expect the penalty term will punish the complex model and make sure the final model after training is simple,

why: since overfitting the training data will do bad for test prediction.